Review On The Spring Operating System

Yukui Ye

Spring operating system is a new OS architecture , designed for a network world, It followed the idea which aims to develop a highly modular, extensible, microkernel-based, distributed, object-oriented operating system. Specifically features include microkernel design, modularity with IDL interfaces, and improved security.

The Spring operating system implements Interface Definition Language(IDL) and a simple client model for objects, which can be easily extended to use different software components. It enables adding new types of objects without go over the kernel, therefore multiple implements of an objects can share the same interface. Besides, there is only microkernel ran in the kernel mode, Kernel only manage VM, process and IPC. Other servers such as I/O, file system are running on user-level, This makes Spring system to be defined more open, flexible, extensible and efficient.

As IDL defines what software component does, the objects in Spring decides how and where the operation is performed. The objects in different address space compile an IDL interface to produce client-side “stub” which will use subcontract mechanism to pass arguments for the call to access object in other space. Server-side “stub” on server side receive requests and convert these requests into an invocation of the actual object, then return results to the client. Subcontracts define how object references refer to object , and manage the interaction between invoker and object by transmitting a request to remote servers, it also makes use of the door. The door provide a protected entry into servers which enable clients to make cross-address-space object invocations. Simply, the door provides a safe way for referencing objects. The domain which provides an address space for applications are mapped through door tables to the actual door by using a nonforgeable door identifiers. The door identifiers are mapped into network handles when transporting over the network, network proxies forward door invocations between machines. As door is implemented inside the microkernel, It makes easier to provide security. Because client process’s thread reference to the index in the door table of the pointer to the door, therefore a client can always refer to a door related to its object reference.

Another Spring goal is to provide secure access to object, It is called the security model. In order to access to an object, we can check Access Control List(ACL) which indicates who is allowed to access this object at runtime. After we done checking, the server of the object generate an object reference which are capabilities. As discussed before, the object reference would link to a door and then the door points to a front object, which encapsulating information identifying the principal in order to access rights, but there comes an issue related to the capabilities based on ACL. How can we restrict the use of a capability ? Spring find a way to solve this problem, that is for an object manager to bind a reference to a front object rather than the real object. This leads to when the client try to invoke method on the real object, it must go through the new front object which will only allow client to use methods that ACL allows, therefore client must hold its reference for front object ”upgrade” in order to allow access to the desired object. The Spring’s Security Model provide a wide rang of secure mechanisms in a networked environment.